Министерство образования Новосибирской области ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

РАЗРАБОТКА СИСТЕМЫ ОТСЛЕЖИВАНИЯ ПОСЕЩЕНИЯ АУДИТОРИЙ СТУДЕНТАМИ

Пояснительная записка дипломного проекта

НАТКиГ.100900.010.000ПЗ

Выполнил:

студент группы ПР-20.102К

Лазарев Д.А.

2024

**Содержание**

[Введение 3](#_Toc21613)

[1 Исследовательский раздел 5](#_Toc4740)

[1.1 Описание предметной области 5](#_Toc17809)

[1.2 Определение функциональных задач пользователей 7](#_Toc22754)

[1.3 Анализ аналогов и прототипов 11](#_Toc31794)

[1.4 Проектирование информационной системы 14](#_Toc4707)

[1.5 Анализ программных ресурсов, необходимых в работе 14](#_Toc18337)

[2 Технологический раздел 17](#_Toc6119)

[2.1 Структура базы данных 17](#_Toc4286)

[2.2 Разработка форм выходных документов 19](#_Toc31795)

[2.3 Разработка структуры программного продукта 20](#_Toc16154)

[2.4 Описание процедур и функций 24](#_Toc4349)

[3 Инструкции пользователям системы 26](#_Toc3131)

[4 Тестирование программного продукта 40](#_Toc17855)

[4.1 Выбор стратегии тестирования 40](#_Toc23982)

[4.2 Разработка сценариев тестирования 40](#_Toc13786)

[Заключение 42](#_Toc17689)

[Библиография 43](#_Toc10505)

## Введение

В современном образовании эффективное управление учебным процессом и контроль посещаемости студентов играют ключевую роль в обеспечении качественного образования и достижении учебных целей. Однако, традиционные методы отслеживания посещаемости часто оказываются неэффективными и трудоемкими как для преподавателей, так и для администрации учебных заведений.

В этом контексте разработка приложения для отслеживания посещаемости студентов становится актуальной и востребованной задачей. Переход к цифровизации и использование современных информационных технологий позволяют создать инновационный инструмент, который сможет упростить и автоматизировать процесс учета посещаемости, обеспечивая при этом надежность, точность и удобство использования.

Цель данного дипломного проекта состоит в разработке и реализации такого приложения, которое позволит эффективно отслеживать посещаемость студентов в учебных заведениях. В рамках работы будут проанализированы существующие подходы к отслеживанию посещаемости, выявлены основные требования к приложению, разработана его структура и интерфейс, а также проведена оценка эффективности и возможных преимуществ от внедрения данного решения в учебный процесс.

Результаты этой работы могут быть полезны как для преподавателей и администрации учебных заведений, так и для самих студентов, способствуя оптимизации учебного процесса, повышению его качества и обеспечению более эффективного контроля за посещаемостью.

Информационная система (далее – ИС) состоит из трех частей:

* база данных;
* серверная часть;
* десктопное приложение.

Десктопное приложение планируется использовать как основной инструмент для управления информационной системой, оно позволит полностью отображать информацию о перемещении студентов внутри стен колледжа, такую как: точное время и дату входа и выхода в аудиторию, сам номер аудитории. Также преподавателю будут доступны инструменты для удобного поиска студента и генерации отчетности, а также интуитивно понятный интерфейс для использования всего функционала приложения.

Десктопное приложение также выступает в роли посредника между базой данных и пользователем, предоставляющим функционал для взаимодействия клиента с ИС. Десктопное приложение должно обрабатывать все входные и выходные данные, которые поступают в системы путем QR кода. Студенты не могут как то пользоваться информационной системой и использовать её функционал, а лишь могут отправлять запросы косвенным способом, а именно с помощью QR кодов, заранее подготовленным преподавателям, воспользовавшимися инструментов внутри десктопного приложения.

Актуальность разрабатываемого программного продукта заключается в том, что он предназначен для замены используемых бумажных журналов, упрощения создания отчетной документации.

Можно выделить следующие задачи дипломного проекта:

* исследование предметной области;
* выбор средств разработки;
* разработка дизайн-проекта;
* разработка программной части;
* разработка базы данных;
* документирование.

## 1 Исследовательский раздел

## Описание предметной области

Предметная область отслеживания посещаемости студентов в учебных заведениях является одной из ключевых сфер управления учебным процессом. Её значимость обусловлена не только тем, что регулярное посещение занятий способствует более эффективному усвоению учебного материала и повышению успеваемости, но и тем, что это позволяет преподавателям и администрации учебных заведений оценить уровень активности студентов, выявить проблемные ситуации и своевременно предоставить необходимую поддержку.

Традиционные методы учета посещаемости, такие как ручное ведение журналов преподавателями, часто оказываются недостаточно эффективными и подвержены ошибкам. Этот процесс также требует значительных временных затрат на обработку данных, что может отвлекать преподавателей от основной образовательной деятельности.

Однако в настоящее время с развитием информационных технологий и цифровизации учебного процесса появляются новые возможности для автоматизации и улучшения процесса отслеживания посещаемости студентов. Создание приложения, позволяющего студентам и преподавателям эффективно отслеживать посещаемость занятий, становится все более актуальным.

Заинтересованными сторонами в данной предметной области являются преподаватели, администрация учебных заведений, студенты и их родители. Для них важно иметь доступ к достоверной информации о посещении занятий и возможность эффективно управлять этим процессом. Также необходимо учитывать вопросы приватности данных, безопасности и удобства использования приложения.

Обоснование актуальности проекта по созданию приложения для отслеживания посещаемости студентов в учебных заведениях основывается на нескольких ключевых аспектах.

Регулярное посещение занятий является ключевым элементом успешного обучения. Отслеживание посещаемости помогает образовательным учреждениям и преподавателям оценить уровень учебной активности студентов, выявить проблемные ситуации и своевременно оказать необходимую поддержку.

Приложение для отслеживания посещаемости может значительно упростить и автоматизировать процесс регистрации студентов на занятиях. Это позволит преподавателям и администрации учебных заведений сосредоточиться на более важных аспектах образовательного процесса.

В условиях быстрого развития информационных технологий и мобильных приложений использование современных технологий становится неотъемлемой частью образовательного процесса. Приложения для отслеживания посещаемости представляют собой пример инновационного подхода к управлению образованием.

Приложение для отслеживания посещаемости также может способствовать улучшению коммуникации между преподавателями и студентами, предоставляя им информацию о текущем состоянии посещаемости и связанных с этим вопросах.

Таким образом, создание приложения для отслеживания посещаемости студентов является актуальным и востребованным проектом, который может значительно облегчить и оптимизировать управление учебным процессом, повысить его эффективность и поддержать активное участие студентов в образовательном процессе.

## Определение функциональных задач пользователей

Серверная часть ИС будет находиться на удалённом сервере частной компании, с которой учебное заведение заключит договор, все же фотоизображения будут храниться на каждом компьютере индивидуально. Сотрудник может взаимодействовать с серверной частью в любое время.

Серверная часть обладает следующим набором функций:

* просмотр записей посещений, включая поиск по имени студента и фильтрацию по дате посещений и по группе студента;
* добавление студента;
* формирование отчета за определенный период времени индивидуально для каждого студента в формате Excel;
* отображение подробной информации о студенте;

Разработка мобильного приложения для ИС не подразумевается, т.к. большого функционала в ней не планируется, вместо этого студенту понадобится любое приложение или встроенное приложение для сканирования QR кода.

Следовательно, мобильная часть ИС обладает следующим набором функций:

* Сканирования QR кода и передачи набора данных о студенте.

На рисунке 1 представлена диаграмма прецедентов для ИС.

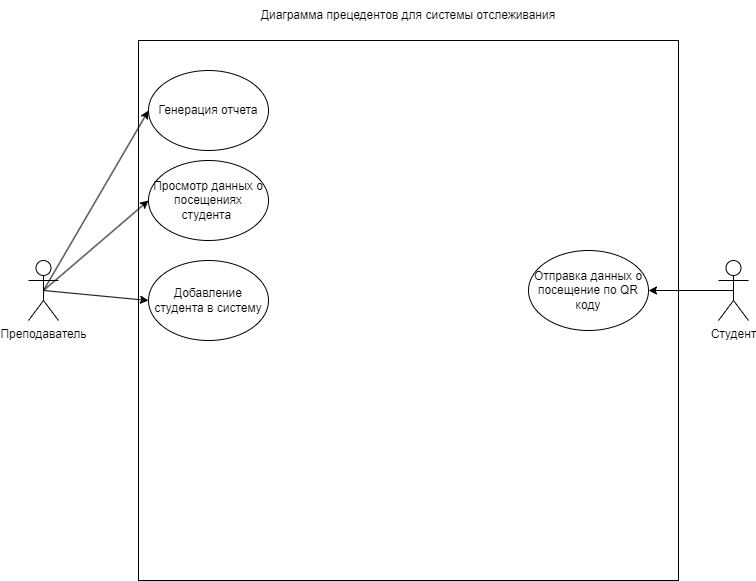


Рисунок 1 – Диаграмма прецедентов ИС

В системе участвуют два актера. Преподаватель, взаимодействующий с серверной частью ИС и студент, который взаимодействует с веб частью.

Спецификации прецедентов указаны в таблицах 1–10, альтернативные потоки обозначены A*n*, где *n* – номер потока.

Таблица 1 – Спецификация для прецедента «Генерация отчета»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |
| Наименование прецедента | Генерация отчета |
| Краткое описание | Позволяет преподавателю сгенерировать отчет о студенте |
| Действующие лица | Преподаватель |
| Предусловия | Система запущена, студент добавлен в систему |
| Основной поток | 1 Система выводит в предложенном меню список студентов по которым необходимо сделать отчет.  2 Сотрудник заполняет необходимые поля и нажимает кнопку «Отчет»  А1 Обязательные поля не заполнены |

Продолжение таблицы 1

|  |  |
| --- | --- |
| 1 | 2 |
| Альтернативный поток | А1 Обязательные поля не заполнены  1 Система информирует пользователя о необходимости  заполнить определенные поля  2 Поток возвращается на 2 этап основного потока |
| Постусловие | Результатом будет генерация отчета о студенте |

В таблице 2 указана спецификация для прецедента «Просмотр данных о посещениях студента».

Таблица 2 – Спецификация для прецедента «Просмотр данных о посещениях студента»

|  |  |
| --- | --- |
| Раздел | Описание |
| Наименование прецедента | Просмотр данных о посещениях студента |
| Краткое описание | Позволяет преподавателю просматривать данные о посещениях студента |
| Действующие лица | Преподаватель |
| Предусловия | У системы есть доступ к интернету |
| Основной поток | 1 Система при запуске открывает окно с выбором студента и этажа  А1 Не выбран студент или этаж  2 Система динамически обновляет информацию о номере аудитории и информацию о посещении |
| Альтернативный поток | А1 Не выбран студент или этаж  1 Система предложит выбрать студента и этаж  2 Поток возвращается на 1 этап основного потока |
| Постусловие | Результатом будет просмотр информации о студенте |

В таблице 3 указана спецификация для прецедента «Добавление студента в систему».

Таблица 3 – Спецификация для прецедента «Добавление студента в систему»

|  |  |
| --- | --- |
| Раздел | Описание |
| Наименование прецедента | Добавление студента в систему |
| Краткое описание | Позволяет преподавателю добавить нового студента |
| Действующие лица | Преподаватель |
| Предусловия | Система запущена |
| Основной поток | 1 Система открывает окно с полями для ввода  2 Сотрудник заполняем необходимые поля и нажимает кнопку «Добавить»  А1 Обязательные поля не заполнены |
| Альтернативный поток | А1 Обязательные поля не заполнены  1 Система информирует пользователя о необходимости  заполнить обязательные поля  2 Поток возвращается на 2 этап основного потока |

В таблице 4 указана спецификация для прецедента «Отправка данных о посещении по QR коду».

Таблица 4 – Спецификация для прецедента «Отправка данных о посещении по QR коду»

|  |  |
| --- | --- |
| Раздел | Описание |
| Наименование прецедента | Отправка данных о посещении по QR коду |
| Краткое описание | Позволяет студенту передавать данные о посещении по QR коду |
| Действующие лица | Студент |
| Предусловия | Система запущена, у студента есть доступ в интернет |
| Основной поток | 1 Студент сканирует QR код и переходит по ссылке  А1 QR код не определен  2 Система записывает информацию о посещении аудитории |
| Альтернативный поток | А1 QR код не определен  1 Система выдает сообщение что данные не были добавлены |
| Постусловие | Результатом будет запись информации о посещении в базу данных |

## 1.3 Анализ аналогов и прототипов

«МОЙ КЛАСС» — коммерческий продукт представленный в виде CRM системы, предназначенный для автоматизации учебного процесса. Сервис реализует функции обработки данных о посещениях, управления учебного процесса, осуществления процесса коммуникации между учниками и преподавателями.

«МОЙ КЛАСС» предоставляет большое количество возможностей управления учебным процессом, таких как:

* доступ к материалам занятий;
* оповещение и рассылка;
* отчеты и аналитика;
* интеграция с мессенджерами.

«МОЙ КЛАСС» предоставляет удобный интерфейс обслуживания для любого компьютера или смартфона. Интерфейс контроля посещаемости системы представлен на рисунке 2

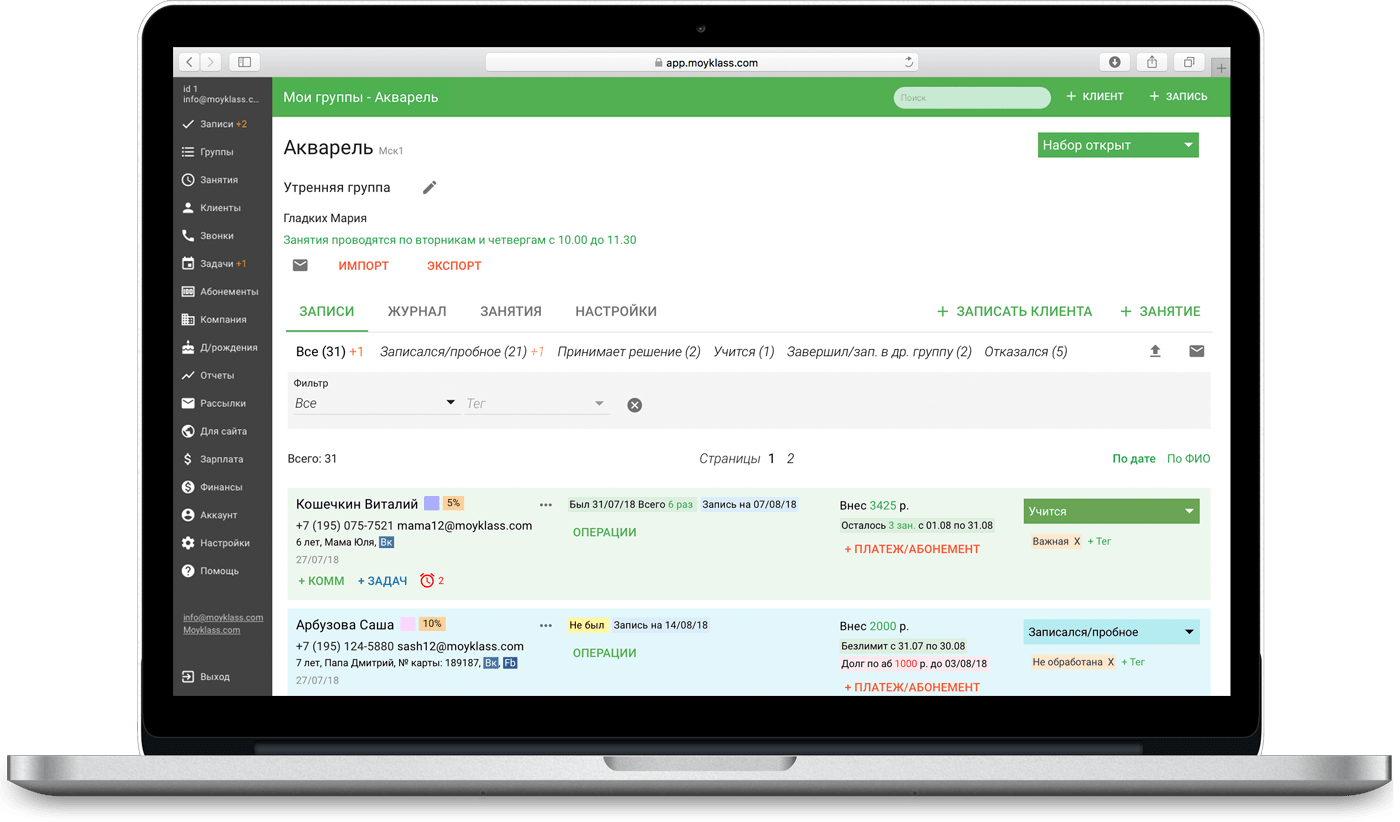


Рисунок 2 – Интерфейс контроля посещаемости системы «МОЙ КЛАСС»

«Дневник.ру» – это веб приложение, включающее все функции электронного дневника. Сервис содержит основные функции: просмотр расписания, оценок, информации об учащемся. В программе можно установить коммуникацию между преподавателем и учащемся путем встроенного чата. Также в системе предусмотрен собственный файлообменник.

Основные функции, предоставляемые системой:

* просмотр посещаемости;
* широкий функционал как учащегося, так и для преподавателя;
* оценивание и посещаемости;
* обмен данными большинства форматов;
* печать отчетности об учебном процессе.

На рисунке 3 представлен интерфейс преподавателя в сервисе «Дневник.ру»



Рисунок 3 – Интерфейс преподавателя

В таблице 5 представлено сравнение двух представленных систем по контролю учебного процесса

Таблица 5 – Сравнение прототипов

|  |  |  |
| --- | --- | --- |
| Критерии\ПО | МОЙ КЛАСС | Дневник.ру |
| Функциональность | Просмотр посещаемости студента, с учетом времени прихода и ухода; Генерация графиков по отчетам; Функции продаж; Функция расчета зарплат сотрудникам;  Онлайн трансляции. | Расписание уроков; Электронный журнал; Управление домашними заданиями; Мобильный журнал. |
| Удобство | Личный кабинет ученика и преподавателя в мобильном приложении; Интеграция API; Календарь занятий и группы; Занятия онлайн и домашние задания; | Мобильное приложение для учеников и преподавателей. |
| Безопасность | Круглосуточная поддержка; Резервное копирование данных; Шифрование данных; Регулярные обновления | Закрытия система с жесткой регистрацией, регулируемая законом; Дата-центры Selectel соответствуют международному стандарту Tier III и требованиям безопасности банков, платежных систем и предприятий электронной коммерции PCI DSS. Хранение персональных данных и их передача по сетям общего пользования полностью соответствует требованиям ФЗ № 152 «О персональных данных»; Контент фильтры |
| Стабильность | Хостинг в России; Круглосуточное обслуживание и проверка серверов | Данные хрнаятся на одном из лучших дата-центров России. Круглосуточное обслуживание. |
| Доступность | Предоставляется бесплатное обслуживание, с ограничением до 5 групп. Пробный период 7 дней. Разнообразие тарифов для разных целей. | Сервис не взимает платы за пользования с преподавателей и учеников. Сложная регистрация учебного заведения в системе со множеством нюансов. |

Все данные о прототипах, представленные в таблице, взяты с официальных сайтов о данной продукции.

## 1.4 Проектирование информационной системы

На рисунке 4 представлена диаграмма с общим функционалом визуального интерфейса серверной части ИС.

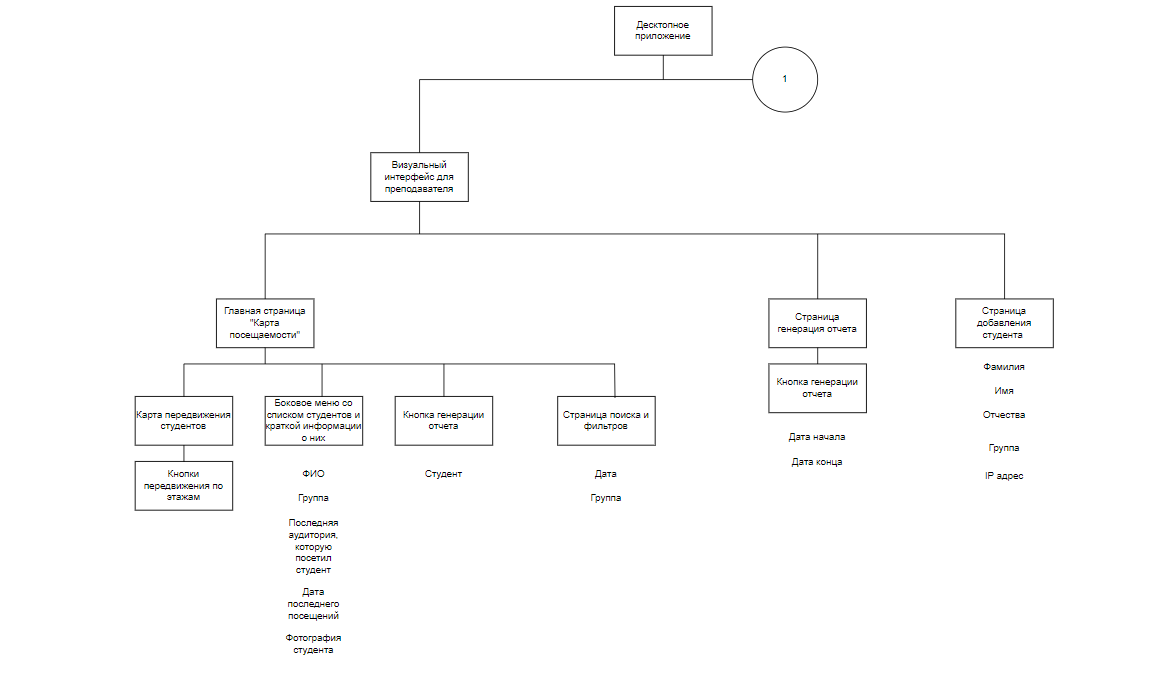


Рисунок 4 – Общий функционал визуального интерфейса серверной части

## 1.5 Анализ программных ресурсов, необходимых в работе

Для разработки базы данных было решено использовать реляционную систему управления базами данных (далее – СУБД) – «phpMyAdmin». Данная система является хорошим решением для малых и средних приложений, а также свободно распространяется по лицензии «GPL» при условии, что программный код разрабатываемого продукта является открытым и свободным в использовании.

Данное СУБД обладает графическим интерфейсом, который позволяет наиболее удобно разрабатывать и конфигурировать базу данных – «MySQL». Для работы с базами данных также используется язык SQL, который изучался в колледже. Описанные характеристики вполне доказывают обоснованность использования данной системы.

Для разработки десктопной части информационной системы было решено использовать «NET.Framework» – технологию для создания десктопных приложений на платформе .NET. Данная технология развивается компанией Microsoft, в качестве языка программирования может использоваться C#.

Данная технология позволяет разрабатывать логическую часть на языке C#, который является основным языком разработки в колледже, и верстку визуальных представлений на языке XAML, что предоставляет удобство при создании функций, таблиц и элементов управления.

Отдельно можно выделить следующие особенности ASP.NET Core:

* работа поверх платформы .NET и полное задействование ее функционала;
* кроссплатформенный фреймворк, приложения на котором могут быть развернуты на всех основных популярных информационных системах: Windows, Mac OS, Linux;
* фреймворк является модульным, что позволяет загружать все необходимые компоненты веб-приложения как отдельные модули через пакетный менеджер NuGet;
* в качестве инструментария разработки можно использовать такую среду разработки с богатым функционалом, как Visual Studio.

Выбор данной технологии обусловлен ее растущей популярностью, удобностью применяемых шаблонов проектирования и положительными особенностями, описанными выше.

Для обработки некоторых функций на стороне клиента предполагается использовать фреймворк JavaScript – JQuerry, который представляет набор функций JavaScript и фокусируется на взаимодействии JavaScript и C#.

Можно заключить, что выбор вышеописанных технологий обусловлен не только их растущей популярностью и востребованностью в мире, но также и удобностью использования – все части информационной системы, за исключением базы данных, разрабатываются в одной среде разработки VisualStudio, их логика разрабатывается на языке C# поверх платформы .NET, что позволит использовать различные дополнительные библиотеки как, например Entity Framework, которая позволяет облегчить взаимодействие с базой данных.

## 2 Технологический раздел

## 2.1 Структура базы данных

База данных включает в себя 4 сущностей. На рисунке 5 представлена ER–диаграмма базы данных.

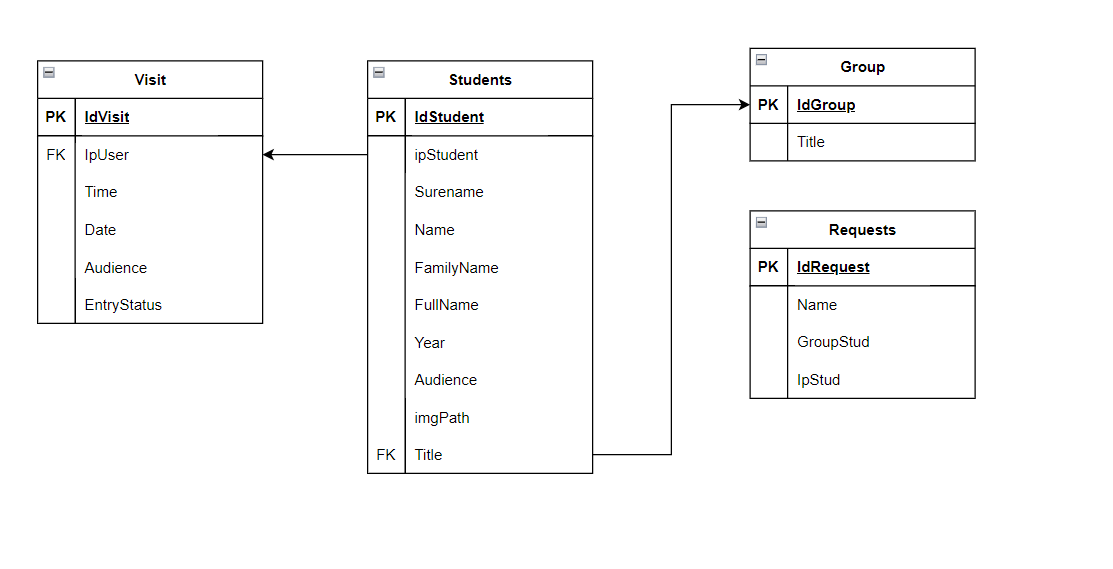


Рисунок 5 – ER–диаграмма базы данных

Сущность Students предназначена для хранения в базе данных информации о студентах.

Таблица 6 – Сущность Students

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| idStudent | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| ipStudent | Nvarchar | No | IP студент |
| Surename | Nvarchar | No | Фамилия студента |
| Name | Nvarchar | No | Имя студента |
| FamilyName | Nvarchar | Yes | Фамилия студента |
| FullName | Nvarchar | No | Поле включающее фамилию, имя отчество |
| Year | Int | No | Год обучения |
| Audience | Int | No | Аудитория в которой студент находился в последний раз |
| imgPath | Nvarchar | No | Путь к аватару студенту |
| Title | Int | No | Внешний ключ к наименованию группы |

Сущность Visit предназначена для хранения в базе данных информации о посещениях студентов.

Таблица 7 – Сущность Visit

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| IdVisit | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| IpUser | Nvarchar | No | IP студента, внешний ключ |
| Time | Time | Yes | Время посещения |
| Date | Date | No | Дата посещения |
| Audience | Int | No | Аудитория |
| EntryStatus | Bool | No | Статус входа/выхода |

Сущность Requests предназначена для хранения информации о запросах для регистрации.

Таблица 8 – Сущность Requests

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| IdRequest | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Name | Nvarchar | No | ФИО студента |
| GroupStud | Nvarchar | No | Группа студента |
| IpStud | Nvarchar | No | IP адрес студента для регистрации |

Сущность Group предназначена для хранения информации о группах.

Таблица 9 – Сущность Group

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип данных | Необязательное поле | Примечание |
| IdGroup | Int | No | Идентификатор таблицы. Автоинкрементный, первичный ключ |
| Title | Nvarchar | No | Наименование группы |

## 2.2 Разработка форм выходных документов

В ИС в качестве выходного документа можно выделить автоматически генерируемый отчетный документ для будущего хранений и использования за пределами аудитории.

Данный документ создается в дестопной части ИС. Для его генерации преподавателю необходимо указать период времени, за который необходимо собрать данные и студента. На рисунке 6 показано окно выбора временного промежутка на странице работы с документацией.

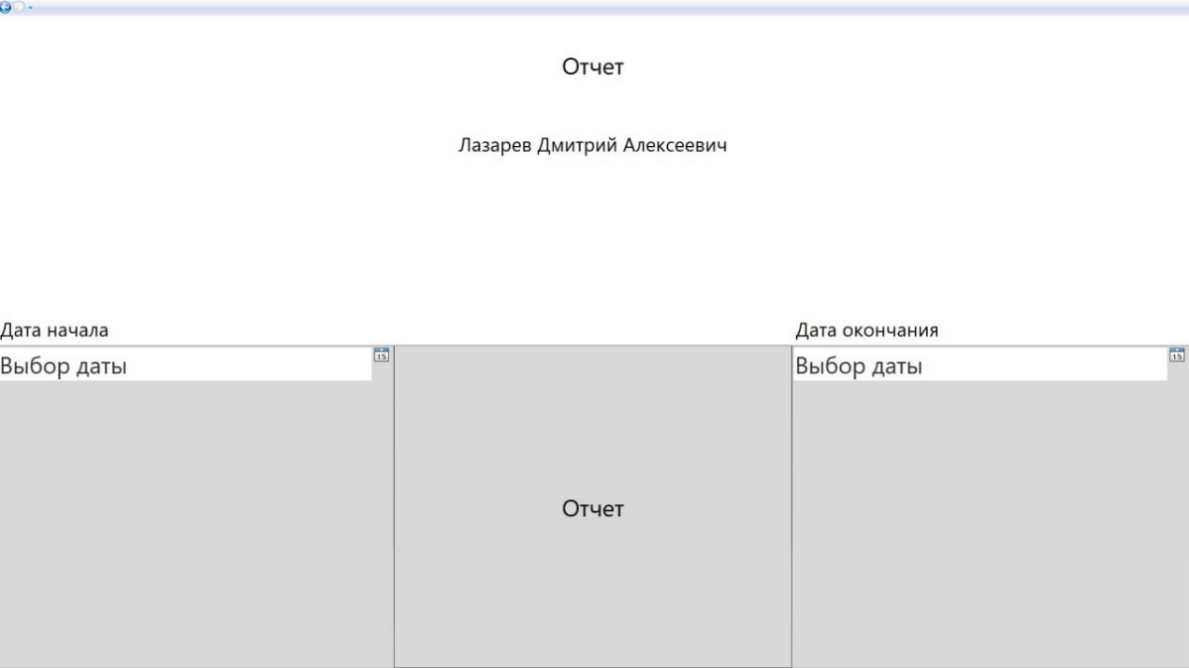


Рисунок 6 – Выбор временного промежутка для генерации отчета

Сам же метод генерации документа получает в качестве входных параметров две переменные – дату старта промежутка и дату конца промежутка. Если выбрана одна дата, то переменные равны между собой и отчет генерируется за один день. После указания дат система собирает необходимые данные из базы данных и передает их в метод генерации документа.

Отчетный документ включает в себя такие параметры, как:

* ФИО студента;
* аудиторий;
* время;
* дата;
* состояние.

На рисунке 7 представлен образец содержания отчетного документа.

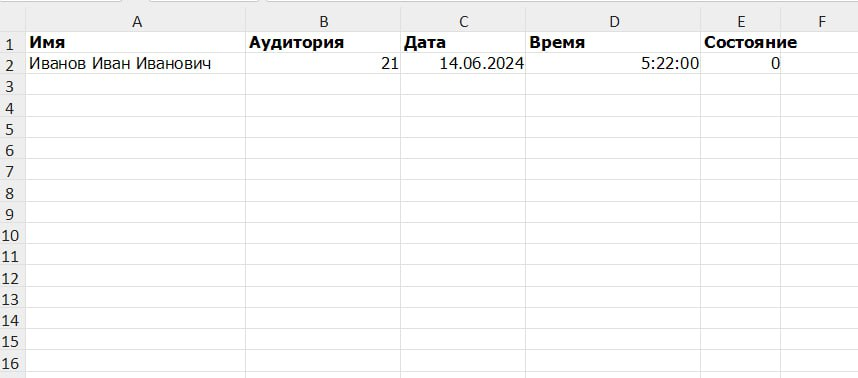


Рисунок 7 – Образец содержания отчетного документа

Отчет является внешним и позволяет получить основную информацию о посещениях студентах внутри колледжа.

## 2.3 Разработка структуры программного продукта

Для реализации архитектуры серверной части ИС было решено использовать шаблон проектирования MVC –Model–View–Controller. Данный шаблон предполагает разделение пользовательского интерфейса (представления), данных (модели) и логики приложения (контроллера) и помогает добиться разделения задач.

Шаблон MVC имеет следующую логику взаимодействия: запросы направляются в контроллер, который отвечает за работу с моделью и обеспечивает выполнение действий с данными и / или их извлечение. В представлении же отображается конечная страница на основе данных из модели. На рисунке 8 изображена схема работы шаблона MVC.

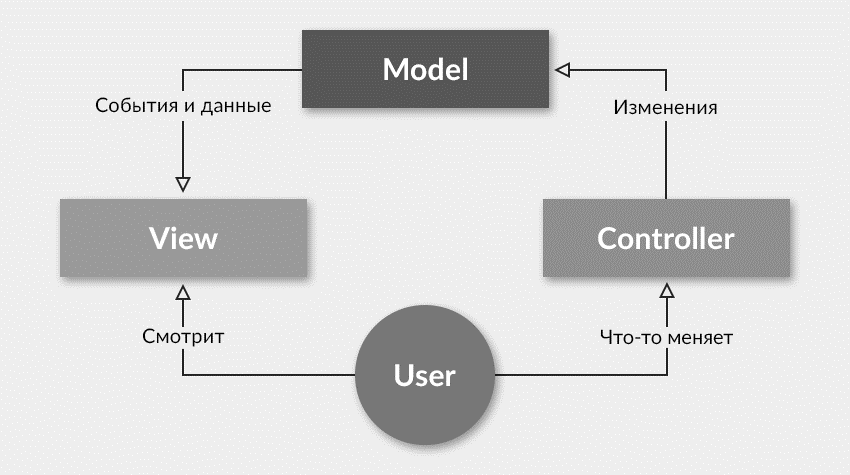


Рисунок 8 – Схема работы шаблона MVC

Взаимодействие между серверной и мобильной частью происходит посредством реализации API-интерфейсов четырех видов: GET, POST, PUT. API интерфейс вида GET позволяет десктопному приложению получать с сервера список студентов и список посещений в текстовом формате обмена данными, основанном на языке JavaScript – JSON. Запросы вида POST и PUT позволяют десктопному приложению передавать определенные данные на сервер, в нашем случае передается информация о перемещениях студентов, осуществленных студентом. При сканировании QR-кода для регистрации посещения на сервер отправляются данные об аудитории, ip студента, время, дата, состояние.

Файлы десктопной части ИС представлены в таблице 11

Таблица 11 – Файлы десктопной части ИС

|  |  |  |
| --- | --- | --- |
| Наименование | Назначение | Примечание |
| 1 | 2 | 3 |
| Properties/AssemblyInfo.cs | Набор сведений о сборке проекта | Содержит список сведений о сборке |
| Entity | Папка, содержащая классы модели данных проекта | Содержит в себе классы модели данных |
| Entity/Class1.cs | Файл содержащий классы и методы модели базы данных | Содержит в себе описание атрибутов сущностей и методов, необходимых для получения и использования данных из запроса |
| FirstStage.xaml | Файл разметки для вывода интерфейса | Содержит разметку для корректного отображения данных о перемещениях студента и интерфейса взаимодействия преподавателя с системой |

Окончание таблицы 11

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Resources | Папка, содержащая различные ресурсы приложения | Содержит в себе различные ресурсы, необходимые для разработки дизайна приложения, такие как шрифты, изображения, иконки и т.п. |
| FirstStage.xaml.cs | Класс модели представления | Содержит в себе различные методы, поля и функции, необходимые для взаимодействия между представлением и моделью с информацией о перемещении и информацией о студентах |
| Report.xaml | Файл разметки для вывода интерфейса | Содержит разметку для корректного отображения интерфейса для генерации отчетности |
| Report.xaml.cs | Класс модели представления | Содержит в себе различные методы, поля и функции, необходимые для генерации отчета |
| Links | Раздел, содержащий зависимости | Содержит все зависимости для корректной работы программы и обеспечения функционала |

## 2.4 Описание процедур и функций

Процедуры и функции, реализованные в серверной части ИС представлены в таблице 12

Таблица 12 – Процедуры и функции серверной части ИС

|  |  |
| --- | --- |
| Имя процедуры | Назначение |
| 1 | 2 |
| Имя функций | Назначение |
| Autho(string Email,string Password) | Функция авторизации пользователя |
| GetGroups() | Получения о всех записанных наименованиях групп |
| GetId() | Получения идентификатора, даты и времени пользователя |
| GetRequest() | Получение всей краткой информации об учащихся |
| GetStudents() | Получения всей полной информации об учащихся |
| GetVisits | Получение всех записей посещения аудиторий |

Процедуры и функции, реализованные в десктопной части ИС представлены в таблице 13

Таблица 13 – Процедуры и функции десктопной части ИС

|  |  |
| --- | --- |
| Имя процедуры | Назначение |
| 1 | 2 |
| FirstStage() | Процедура отвечающая за отрисовку компонентов, сортировку,поиск и добавление студентов |
| public void Check() | Процедура проверяет, присутствовал ли студент на занятиях |
| public string GetTimeVisit(int Audience) | Функция возвращающая время посещения студентов в конкретной аудитории |
| private void btnFirst\_Click(object sender, RoutedEventArgs e) | Процедура заполнения первого этажа |
| private void btnSecond\_Click(object sender, RoutedEventArgs e) | Процедура заполнения второго этажа |
| private void btnThird\_Click(object sender, RoutedEventArgs e) | Процедура заполнения третьего этажа |
| public void Clear() | Процедура отвечающая за очистку формы и перекрашивание кнопки |
| public void Fill(int Floor) | Метод для |
| private void Student\_SelectionChanged(object sender, SelectionChangedEventArgs e) | Процедура отображает информацию о студенте при его выборе |
| public void Info(DateTime dateTime) | Процедура отвечающая за индексацию этажей и заполнение информации о студенте на карте |
| public void RenameFloor(int Floor) | Процедура отвечающая за переименование подписей этажей |
| public void Aligment() | Процедура установки ориентации текстбокса |
| public void AddGroups() | Процедура добавления груп для групбокса |
| private void cbGroups\_SelectionChanged(object sender, SelectionChangedEventArgs e) | Процедура выбора группы |
| private void tbSearch\_SelectionChanged(object sender, RoutedEventArgs e) | Процедура поиска студента |
| private void btnClose\_Click(object sender, RoutedEventArgs e) | Процедура закрытия приложения |
| private void dpDate\_SelectedDateChanged(object sender, SelectionChangedEventArgs e) | Процедура выбора даты посещения |
| private void btnReport\_Click(object sender, RoutedEventArgs e) | Процедура перехода |
| public Report(string ip) | Процедура отвечающая за прорисовку интерфейса на странице |
| private void btnReport\_Click(object sender, RoutedEventArgs e) | Процедура отвечающая за генерацию отчета и файла, создание файла Excel, его последующего заполнения |

## 3 Инструкции пользователям системы

Десктопная часть ИС имеет следующий набор функций и задач:

* просмотр записей посещения студентов на карте;
* поиск студента;
* фильтрация списка студентов;
* переход между этажами;
* переход к экрану создания отчета;
* закрытие приложения;
* генерация отчета;
* добавление студента;

Ниже приведено описание пользовательских операций для выполнения основных задач.

Задача 1: Просмотр записей посещения студентов на карте

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;
* есть доступ к интернету

Подготовительные действия:

* открытие десктопного приложения на устройстве;
* информация о студенте известна пользователю.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 5 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, на правой навигационной панели необходимо выбрать интересующего студент. На рисунке 9 представлена часть главного окна с верхней навигационной панелью.

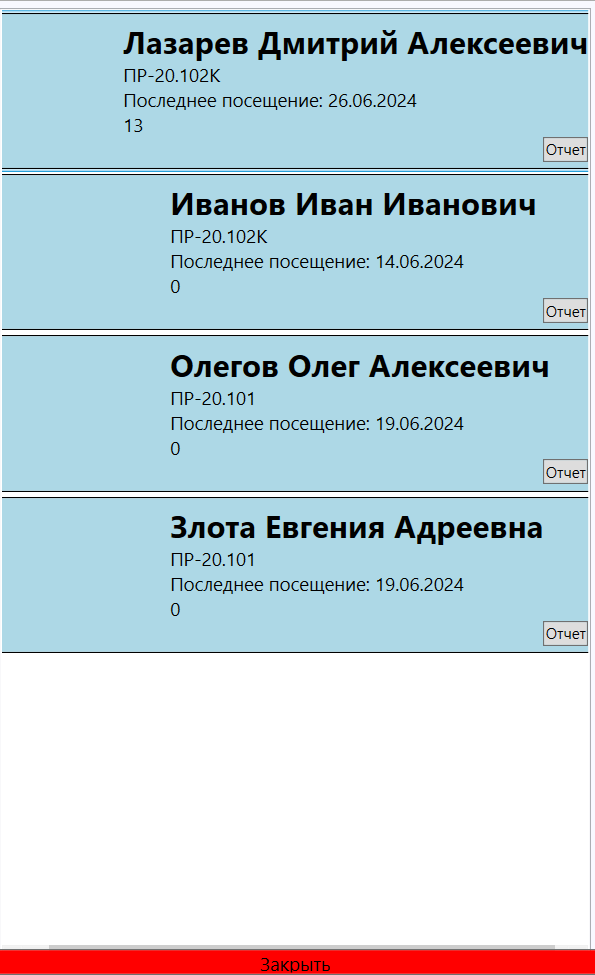


Рисунок 9 – Часть главного окна и вкладка «Просмотр студентов»

1. На верху выбрать дату интересующую преподавателя. На рисунке 10 представлена страница, отображающая перемещения студентов.

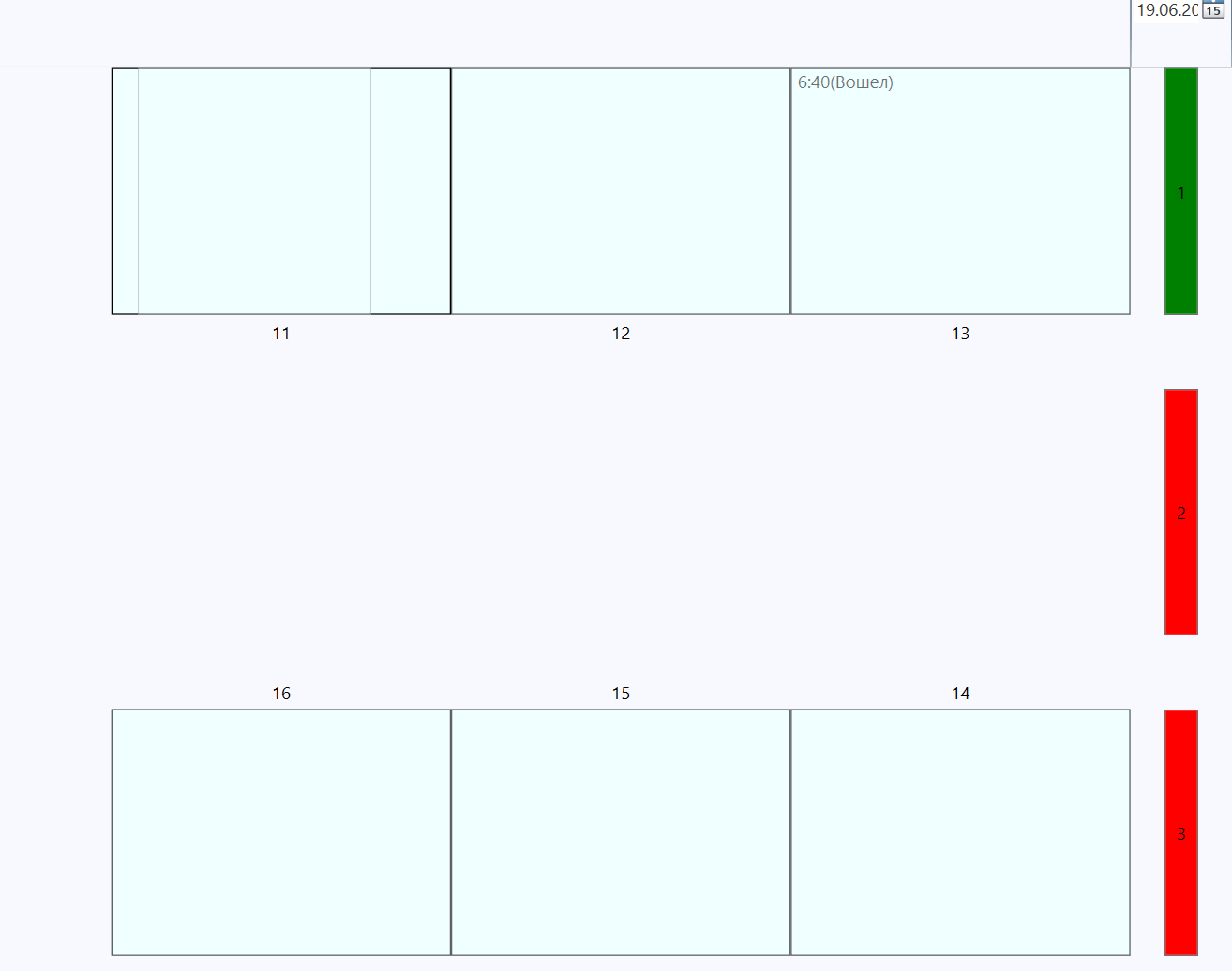


Рисунок 10 – Страница отображения посещений

Задача 2: Поиск студента

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;
* есть доступ к интернету

Подготовительные действия:

* открытие десктопного приложения на устройстве;
* информация о студенте известна пользователю.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 5 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, необходимо выбрать поле поиска. На рисунке 11 показан выбор поля поиска.

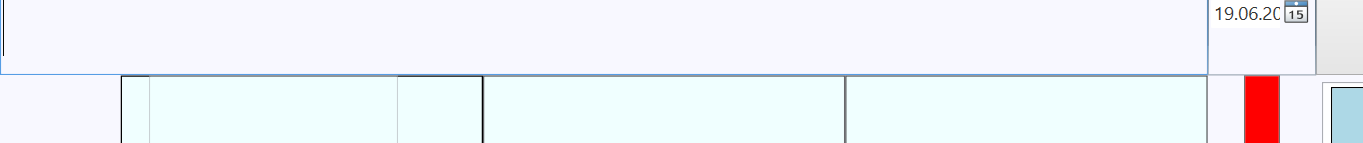


Рисунок 11 – Выбранное поле поиска

1. После выбора необходимо ввести ФИО студента. Заполнение поле поиска представлено на рисунке 12

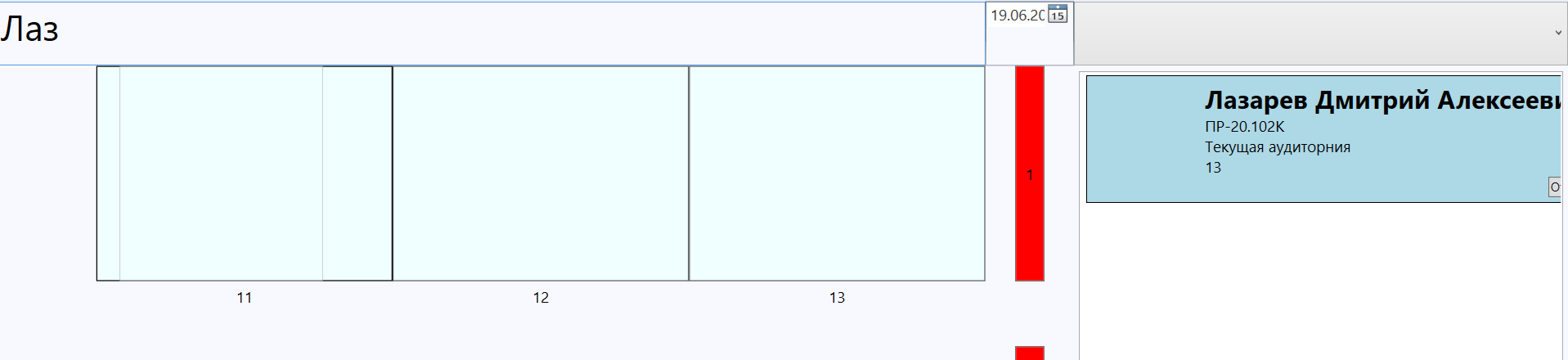


Рисунок 12 – Заполнение поле поиска

1. После заполнения поле поиска необходимо левой кнопкой мыши выделить интересующего студента. Выделение показано на рисунке 13

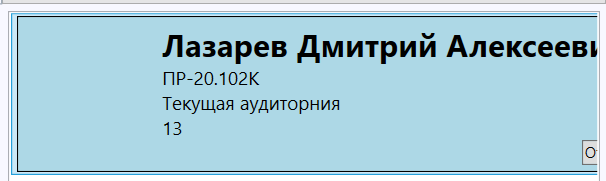


Рисунок 13 – Визуальное выделение студента

Задача 3: Фильтрация списка студентов

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;
* есть доступ к интернету

Подготовительные действия:

* открытие десктопного приложения на устройстве;
* информация о студенте известна пользователю.

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 10 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, в верхнем правом углу необходимо выбрать элемент для отображения списка групп. На рисунке 14 представлено часть главного окна со списком групп.

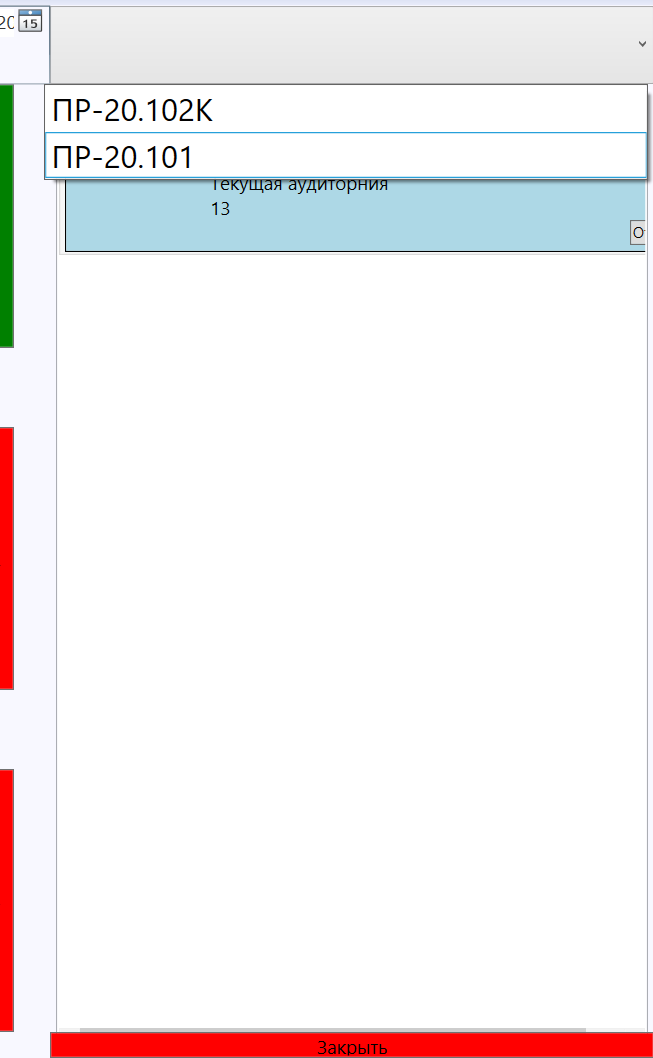


Рисунок 14 – Список групп

1. После клика по группе выведутся все студенты принадлежащие к этой группе. Отфильтрованный список студентов предоставлен на рисунке 15

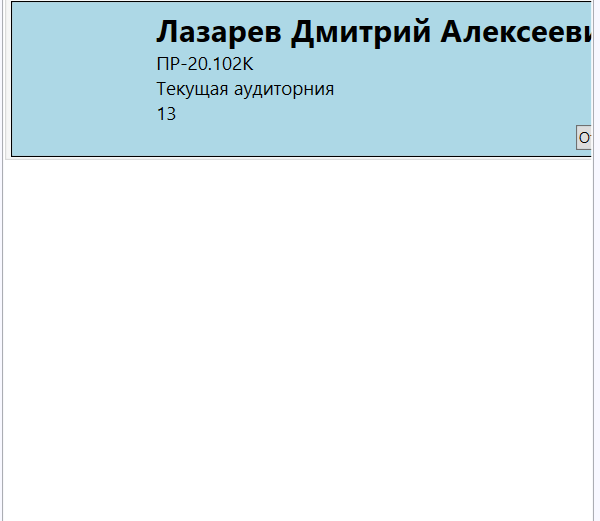


Рисунок 15 – Отфильтрованный список студентов

Задача 4: Переход между этажами

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;

Подготовительные действия:

* открытие десктопного приложения на устройстве;

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 5 секунд.

Основные действия в требуемой последовательности:

1. В левой части экрана необходимо выбрать интересующий этаж, щелкнув по кнопке. Кнопки отображены на рисунке 16

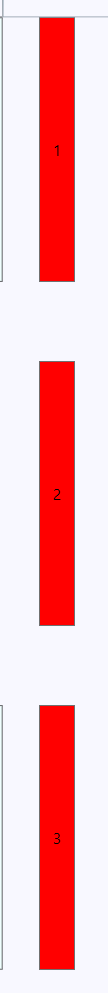


Рисунок 16 – Кнопки выбора этажа

1. После выбора этажа интерфейс карты соответственно изменится. Отображение интерфейса изображено на рисунке 17

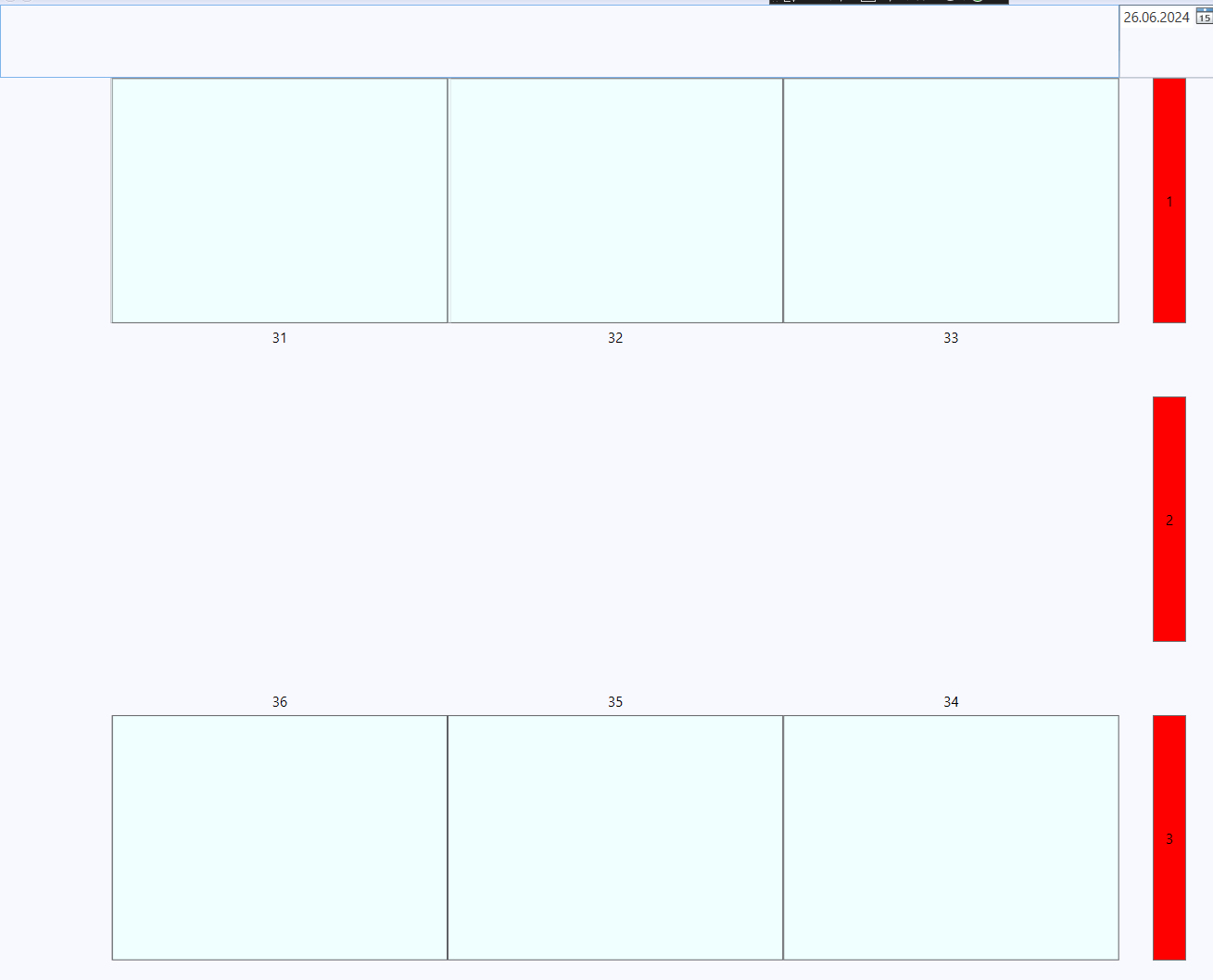


Рисунок 17 – Измененный интерфейс

Задача 5: Переход к экрану отчета

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;
* есть доступ к интернету

Подготовительные действия:

* открытие десктопного приложения на устройстве;
* информация о студенте известна пользователю;

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 40 секунд.

Основные действия в требуемой последовательности:

1. В открывшемся главном окне, в правой части страницы необходимо выбрать студента на которого необходимо составить отчет.
2. У выделенного студента нажать на кнопку «Отчет». Кнопка отчет изображена на рисунке 18

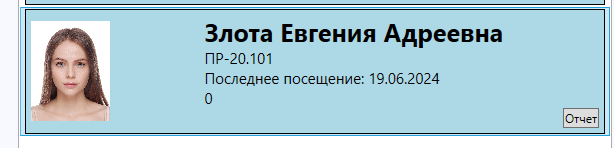


Рисунок 18 – Кнопка отчет на панеле с информацией о студенте

1. У После нажатия произойдет переход на окно составления отчета.

Задача 6: Закрытие приложения

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;

Подготовительные действия:

* открытие десктопного приложения на устройстве;

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 5 секунд.

Основные действия в требуемой последовательности:

1. На главном экране необходимо нажать кнопку «Закрыть». На рисунке 19 изображена кнопка закрытия.



Рисунок 19 – Кнопка закрытия

Задача 7: Генерация отчета

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;
* есть доступ к интернету
* Пользователь находит на странице генерации отчета

Подготовительные действия:

* открытие десктопного приложения на устройстве;
* информация о студенте известна пользователю

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 40 секунд.

Основные действия в требуемой последовательности:

1. Пользователю необходимо выбрать дату начала и дату конца по которым будет происходить генерация отчета. Поля выбра изображены на рисунке 20 и рисунке 21

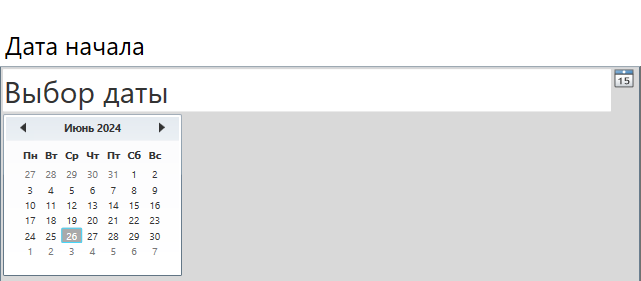


Рисунок 20 – Выбор даты начала

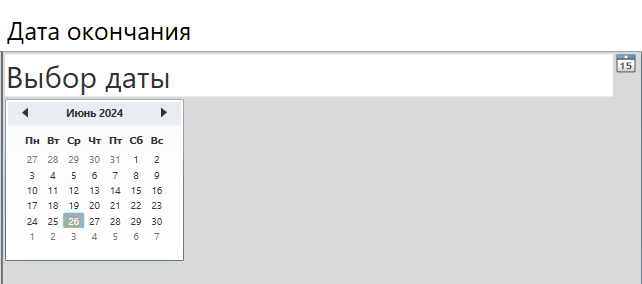


Рисунок 21 – Выбор даты конца

1. Преподавателю необходимо нажать на кнопку для осуществления генерации отчета. Кнопка изображена на рисунке 22

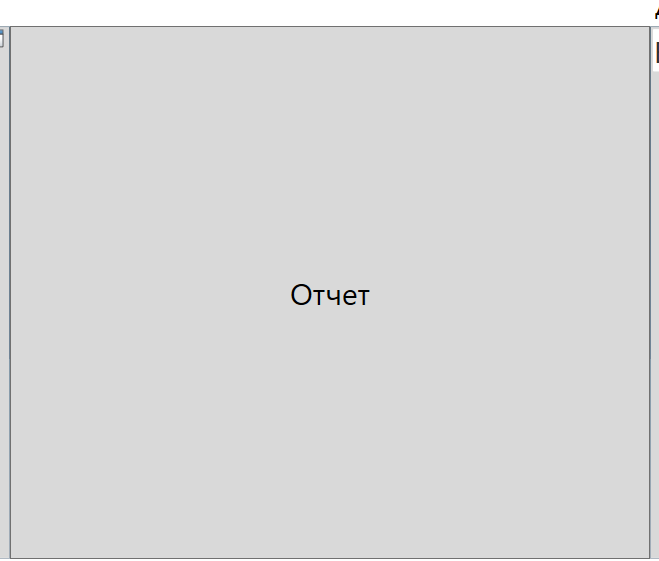


Рисунок 22 – Кнопка генерации отчета

1. После нажатия на кнопку создастся отчет в папке проекта, который можно открыть любой програмой, поддерживающей формат xls.

Задача 8: Добавление студента

Условия, при соблюдении которых возможно выполнение задачи:

* устройство включено;
* десктопное приложение доступно и запущено;
* приложение исправно функционирует;
* есть доступ к интернету
* Пользователь находит на странице генерации добавления студента

Подготовительные действия:

* открытие десктопного приложения на устройстве;
* информация о студенте известна пользователю

Заключительные действия: не требуются.

Ресурсы, расходуемые на операцию: 40 секунд.

Основные действия в требуемой последовательности:

1. Преподавателю необходимо заполнить информацию о студенте. Окно заполнение изображено на рисунке 23

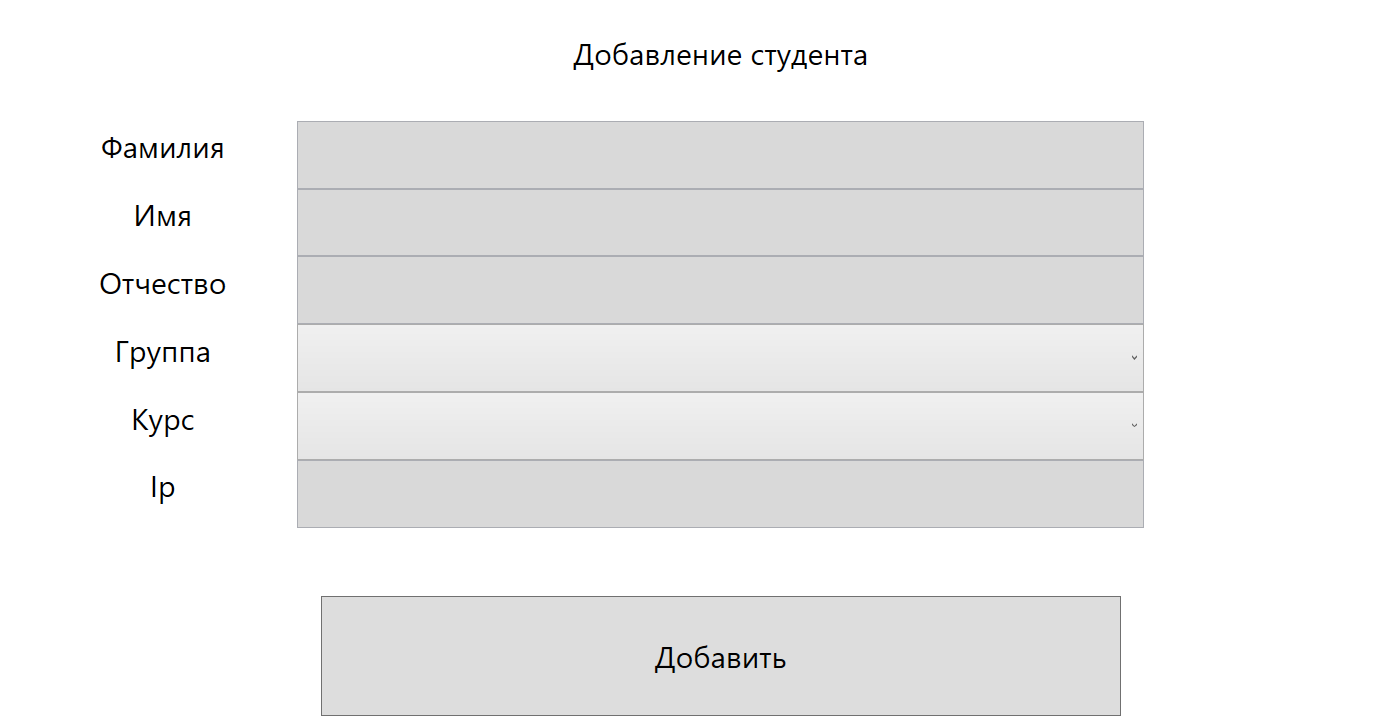


Рисунок 23 – Окно заполнения информации о студенте

1. Преподавателю необходимо нажать на кнопку «Добавить».
2. При успешном добавлении отобразить сообщение об добавлении. Окно изображено на рисунке 24

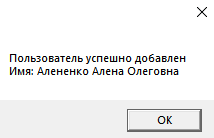


Рисунок 24 – Окно об успешном добавлении

## 4 Тестирование программного продукта



## 4.1 Выбор стратегии тестирования

Метод черного ящика был выбран для оценки эффективности системы, сосредоточенный на проверке её функциональности без учета внутренней архитектуры. Этот подход использует данные, поступающие и исходящие из системы, а также её структуру, не требуя доступа к исходному коду программы. Это позволяет оценить систему с позиции конечного пользователя.

Этот метод критичен для независимой оценки работы программного обеспечения под внешними воздействиями. Он способствует обнаружению возможных проблем и несоответствий между функциональностью и ожиданиями пользователя, что способствует повышению качества продукта и удовлетворенности клиентов.

Тестирование проводится через введение разнообразных входных данных и анализ полученных результатов. Процесс включает разработку тестовых сценариев, основанных на требованиях к функциональности и предполагаемому поведению программы, а затем их выполнение для выявления ошибок, аномалий и отклонений в работе программного обеспечения.

Итоги тестирования могут включать отчёты об обнаруженных дефектах, оценку степени покрытия функциональности тестами, а также предложения по улучшению качества программного продукта.

## 4.2 Разработка сценариев тестирования

Для тестирования функционала были разработаны тест-кейсы, каждый из которых описывает небольшую последовательность действий с одним конкретным результатом. Благодаря тест-кейсам всегда известно, как и что протестировать оптимальным количеством проверок, и не забыть о нюансах, так как записан каждый шаг.

В таблице 14 представлен первый test-case

Таблица 14 – test-case 1

|  |  |
| --- | --- |
| Наименование | Описание |
| Test case # | 1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Получение данных о посещениях студентов |
| Шаги тестирования | 1 Перейти на страницу «Главная»  2 Выбрать студента |
| Данные тестирования | Массив посещений выбранного студента |
| Ожидаемый результат | Вывод посещений в richtextbox |
| Фактический результат | Вывод посещений в richtextbox |

В таблице 15 представлен второй test-case

Таблица 15 – test-case 2

|  |  |
| --- | --- |
| Наименование | Описание |
| Test case # | 2 |
| Приоритет тестирования | Высокий |
| Название тестирования | Получение данных о студентах |
| Шаги тестирования | 1 Перейти на страницу «Главная» |
| Данные тестирования | Получение массива информации о студентах |
| Ожидаемый результат | Вывод данных в listview |
| Фактический результат | Вывод данных в listview |

## Заключение

В результате работы было разработаны серверная и клиентская часть программного продукта, позволяющие просматривать местонахождение студентов на территории колледжа относительно аудиторий.

Серверная часть обеспечивает удобную работу с информацией о объектах, позволяя добавлять новые объекты. Клиентская часть позволяет пользователю быстро получить необходимую информацию.

В ходе выполнения курсового проекта были решены следующие задачи:

* проведен анализ предметной области, определены требования для автоматизации контроля за посещаемостью;
* разработан программный продукт, определяющие его основные модули, функциональные блоки и связи между ними, что позволило логически организовать программное решение и обеспечить взаимодействие между его частями;
* разработан дизайн-макет, включающий в себя интерфейс приложения и элементы управления;

В дальнейшем планируется расширить функционал программного продукта.

## Библиография

Нормативно-правовые акты:

1 ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва: Стандартинформ, 2019. 36 с.

Электронные ресурсы:

1 Metanit.com [Электронный ресурс]. – Руководство по ASP.NET Core MVC. – URL: https://metanit.com/sharp/aspnetmvc/ (дата обращения 12.01.2024)

2 Metanit.com [Электронный ресурс]. – Руководство по MySQL. – URL: https://metanit.com/sql/mysql/ (дата обращения 15.01.2024)

3 Metanit.com [Электронный ресурс]. – Руководство по PHP. – URL :https://metanit.com/php/ (дата обращения 15.01.2024)